Value-at-Risk Evaluation and Density Forecasting

Kevin Sheppard

Today

- Value-at-Risk
  - Model evaluation
- Density Forecasting
- Expected Shortfall

Evaluation of Value-at-Risk Models

- Generalized Mincer-Zarnowitz
$$\mathrm{HIT}_{t+h} = \gamma_0 + \gamma_1 VaR_{t+h|t} + \gamma_2 \mathrm{HIT}_t + \gamma_3 \mathrm{HIT}_{t-1} + \ldots + \gamma_K \mathrm{HIT}_{t-K+1} + \eta_t$$

- Bernoulli
$$I_{[r_t<q_t]} \sim \mathrm{Bernoulli}(\alpha)$$

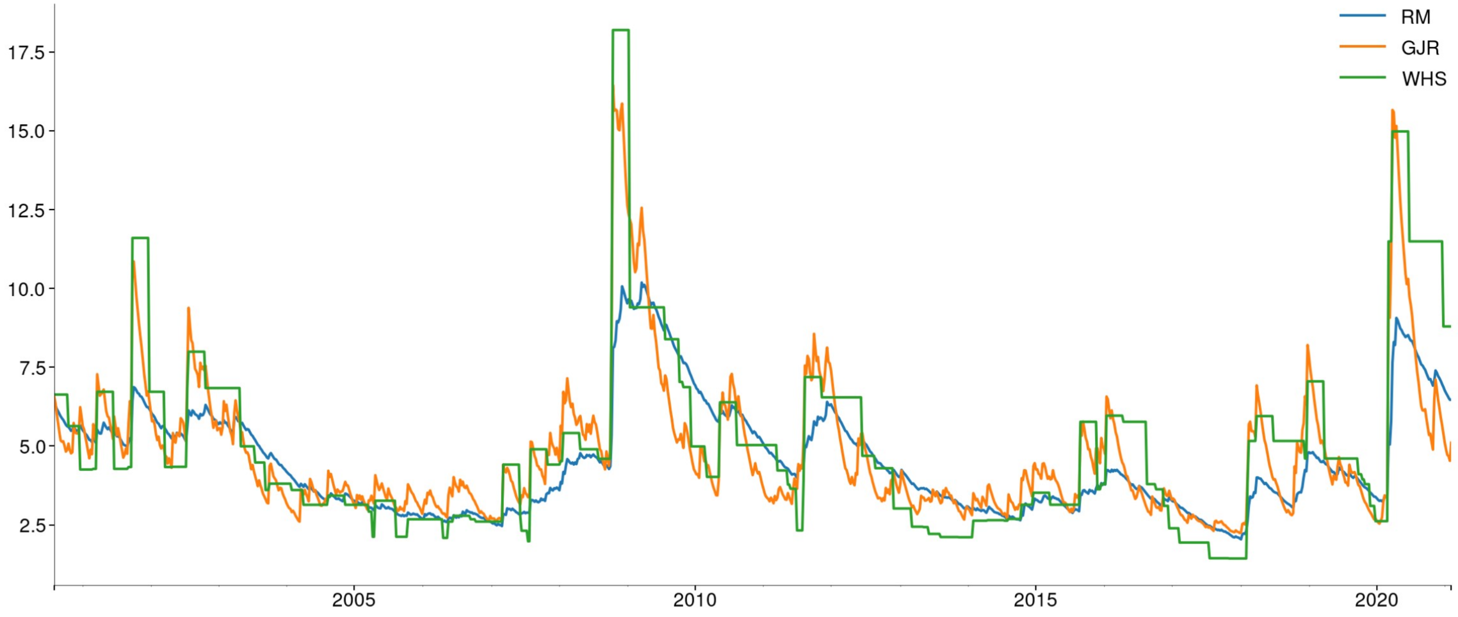
- Christoffersen's Conditional Bernoulli Test
$$I_{[r_t<q_t]} \Big| \, r_{t-1}, q_{t-1} \sim \mathrm{Bernoulli}(\alpha)$$

- Logit/Probit improvement of GMZ
$$I_{[r_t<q_t]} \Big| \, \mathcal{F}_{t-1} \sim \mathrm{Bernoulli}(\alpha)$$

Weekly VaRs for S&P 500

In [4]:

```
plot(weekly_vars)
```



Evaluation: GMZ Regression

- Standard GMZ
$$\mathrm{HIT}_{t+h} = \gamma_0 + \gamma_1 VaR_{t+h|t} + \gamma_2 \mathrm{HIT}_t + \gamma_3 \mathrm{HIT}_{t-1} + \ldots + \gamma_K \mathrm{HIT}_{t-K+1} + \eta_t$$
- Called Dynamic Quantile Regression or HIT Test
- $H_0 : \gamma_j = 0 \; \forall j$

GMZ for Skew t GJR

In [7]:
```
gjr_res = gmz(hits, weekly_vars, "GJR")
summary(gjr_res)
```

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.0133	0.010	1.305	0.192	-0.007	0.033
VaR	-0.0020	0.002	-1.240	0.215	-0.005	0.001
HIT.L1	0.0737	0.054	1.374	0.169	-0.031	0.179
HIT.L2	0.0022	0.034	0.064	0.949	-0.064	0.069
HIT.L3	0.0075	0.031	0.241	0.810	-0.054	0.069

In [8]:
```
joint_test(gjr_res)
```

Out[8]:

	stat	P-value
Joint Test	1.146883	0.332939

GMZ for RiskMetrics

In [9]:
```
rm_res = gmz(hits, weekly_vars, "RM")
summary(rm_res)
```

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.0458	0.015	3.149	0.002	0.017	0.074
VaR	-0.0082	0.002	-3.469	0.001	-0.013	-0.004
HIT.L1	0.0605	0.051	1.184	0.236	-0.040	0.161
HIT.L2	0.0926	0.057	1.637	0.102	-0.018	0.203
HIT.L3	0.0328	0.044	0.740	0.459	-0.054	0.120

In [10]:
```
joint_test(rm_res)
```

Out[10]:

	stat	P-value
Joint Test	5.022963	0.000517

GMZ for WHS

In [11]:

```
whs_res = gmz(hits, weekly_vars, "WHS")
summary(whs_res)
```

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.0325	0.011	2.930	0.003	0.011	0.054
VaR	-0.0050	0.001	-3.570	0.000	-0.008	-0.002
HIT.L1	0.0853	0.053	1.602	0.109	-0.019	0.190
HIT.L2	-0.0117	0.030	-0.392	0.695	-0.070	0.047
HIT.L3	0.0589	0.047	1.254	0.210	-0.033	0.151

In [12]:

```
joint_test(whs_res)
```

Out[12]:

	stat	P-value
Joint Test	4.872636	0.000676

Evaluation: Unconditional Bernoulli Testing

- Likelihood of T exceedences

$$l(\alpha;\widetilde{\mathrm{HIT}}_t) = \sum_{t=1}^{T} \widetilde{\mathrm{HIT}}_t \ln \alpha + \left(1 - \widetilde{\mathrm{HIT}}_t\right) \ln 1 - \alpha$$

- $\widetilde{\mathrm{HIT}}$ is the indicator for a violation
- Easy to conduct a LR test

$$LR = 2(l(\hat{\alpha};\widetilde{\mathrm{HIT}}) - l(\alpha_0;\widetilde{\mathrm{HIT}})) \sim \chi^2_1$$

In [14]:

```
bernoulli_lrs
```

Out[14]:

	alpha	LR	P_value
GJR	0.028945	0.651270	0.419659
RM	0.035481	4.280032	0.038563
WHS	0.033613	2.947361	0.086018

Conditional Bernoulli Test

Christoffersen's test

- Probability of a violation is independent of previous violation
- Leads to conditional Bernoulli distribution
- Has a close form expression
- Key inputs all depend on pairs $I_{[r_t<q_t]}$ and $I_{[r_{t-1}<q_{t-1}]}$

In [16]:
```
christoffersen_lrs
```

Out[16]:

	LR	P_value
GJR	4.000991	0.135268
RM	8.109426	0.017340
WHS	7.424546	0.024422

Evaluation: Using Logit to improve GMZ

- Same model as GMZ
- Exploit structure of exceedence to estimate using MLE
- Requires ensuring conditional probability is always $\in (0, 1)$
  - Logit uses logistic function $\Lambda(\cdot)$
  - Probit uses normal cdf $\Phi(\cdot)$

Logit Results for Skew t

- Null for constant is $\Lambda^{-1}(\alpha) = -3.66$
- All other coefficients are 0 under the null

In [18]:
```
summary(logit_res)
```

	coef	std err	z	P>|z|	[0.025	0.975]
const	-3.2243	0.416	-7.751	0.000	-4.040	-2.409
VaR	-0.0814	0.080	-1.024	0.306	-0.237	0.074
HIT.L1	1.4990	0.647	2.315	0.021	0.230	2.768
HIT.L2	0.1203	1.110	0.108	0.914	-2.056	2.296
HIT.L3	0.2558	0.991	0.258	0.796	-1.686	2.198

In [19]:
```
joint_logit_stat(logit_res)
```

Out[19]:

	stat	p-value
Joint Test	9.267278	0.09886662583981386

Logit Results for RiskMetrics

In [21]:
```
summary(logit_res_rm)
```

	coef	std err	z	P>|z|	[0.025	0.975]
const	-2.0483	0.470	-4.358	0.000	-2.969	-1.127
VaR	-0.3565	0.110	-3.251	0.001	-0.571	-0.142
HIT.L1	1.0894	0.590	1.845	0.065	-0.068	2.246
HIT.L2	1.4879	0.544	2.738	0.006	0.423	2.553
HIT.L3	0.7529	0.658	1.144	0.253	-0.537	2.043

In [22]:
```
joint_logit_stat(logit_res_rm)
```

Out[22]:

	stat	p-value
Joint Test	39.064293	2.3051130840799883e-07

Density Forecasting

- Builds off of ARCH modeling

$$
\begin{aligned}
r_{t+1} &= \mu + \epsilon_{t+1} \\
\sigma^2_{t+1} &= \omega + \gamma\epsilon^2_t + \beta\sigma^2_t \text{ (Any ARCH Process)} \\
\epsilon_{t+1} &= \sigma_{t+1} e_{t+1} \\
e_{t+1} &\overset{iid}{\sim} g(0,1).
\end{aligned}
$$

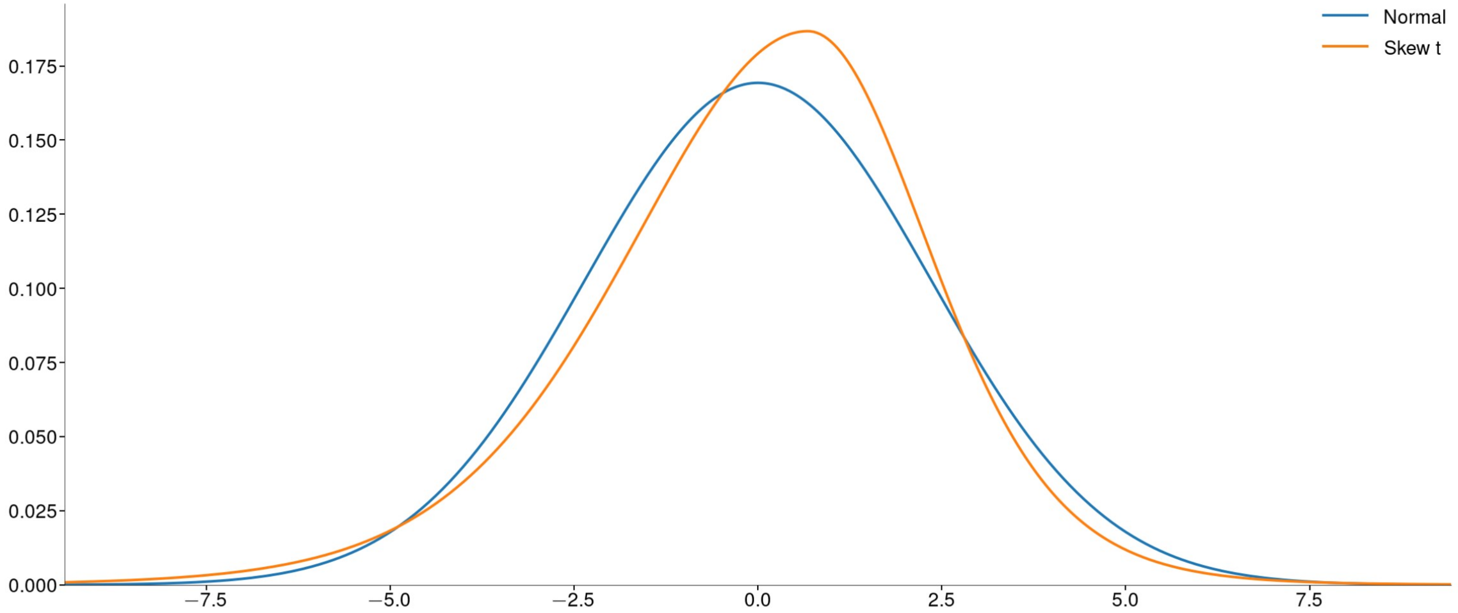
- g is some known distribution, but not necessarily normal
- Density forecast is $g(\mu, \sigma^2_{t+1|t})$
- Flexible through choice of g

Working model

- Normal and Skew t TARCH(1,1,1)
- Weekly S&P 500 Returns

Density Plots: Normal and Skew t

In [24]:
```
xl = density_plots()
```



Kernel Densities

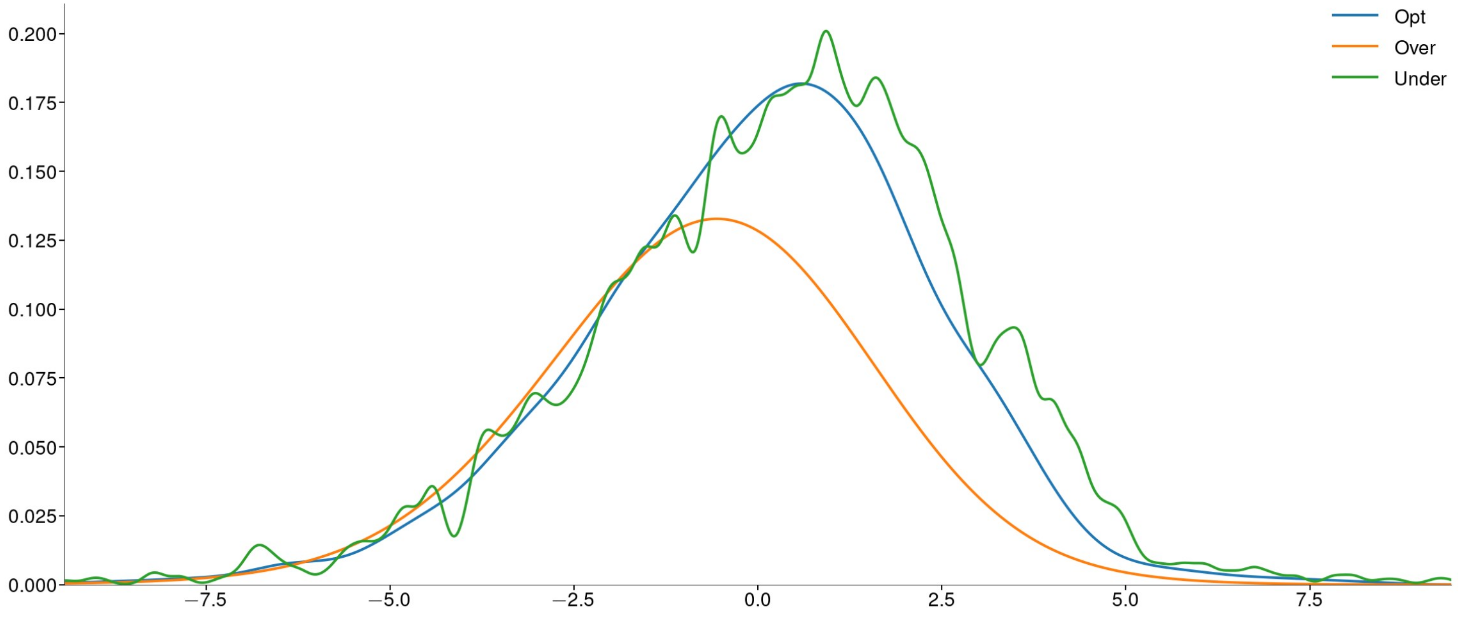
- *Smoothed* densities are more precise than rough estimates

$$g(e) = \frac{1}{Th}\sum_{t=1}^{T} K\left(\frac{\hat{e}_t - e}{h}\right), \quad \hat{e}_t = \frac{y_t - \hat{\mu}_t}{\hat{\sigma}_t} = \frac{\hat{\epsilon}_t}{\hat{\sigma}_t}$$

- Local average of how many $\hat{e}_t$ there are in a small neighborhood of e
- h is the bandwidth
    - Key parameter
    - Bias-variance trade-off

The effect of smoothing

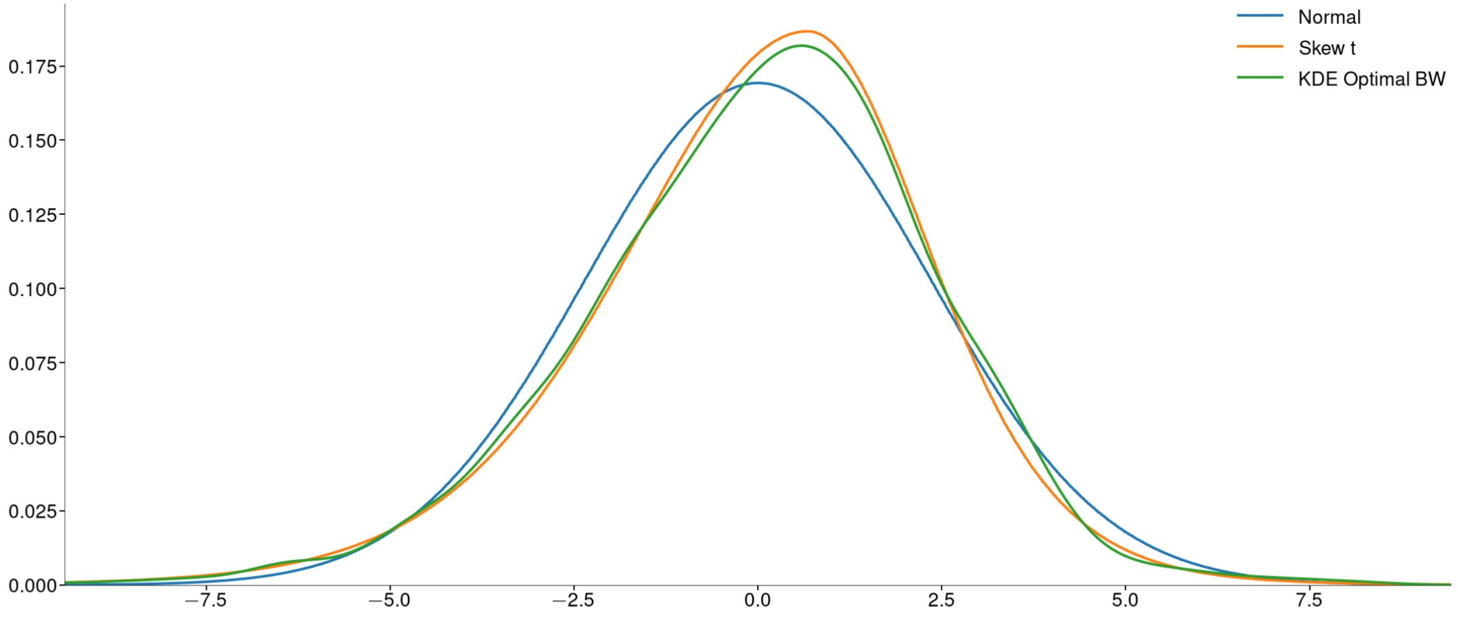
In [26]:
```
kde_plot()
```



Parametric vs KDE Densities

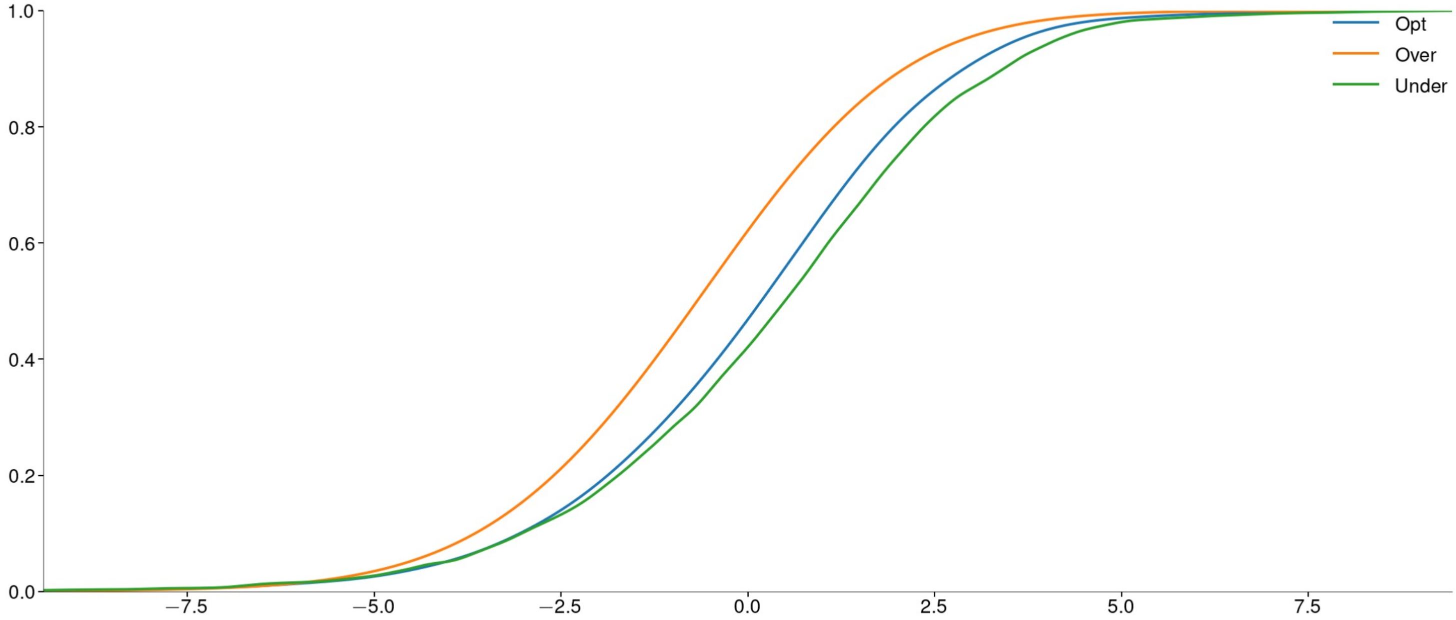
In [28]:

```
compare_densities()
```



Kernel CDFs

In [30]:
```
kernel_cdfs()
```



Multi-step Density Forecasting

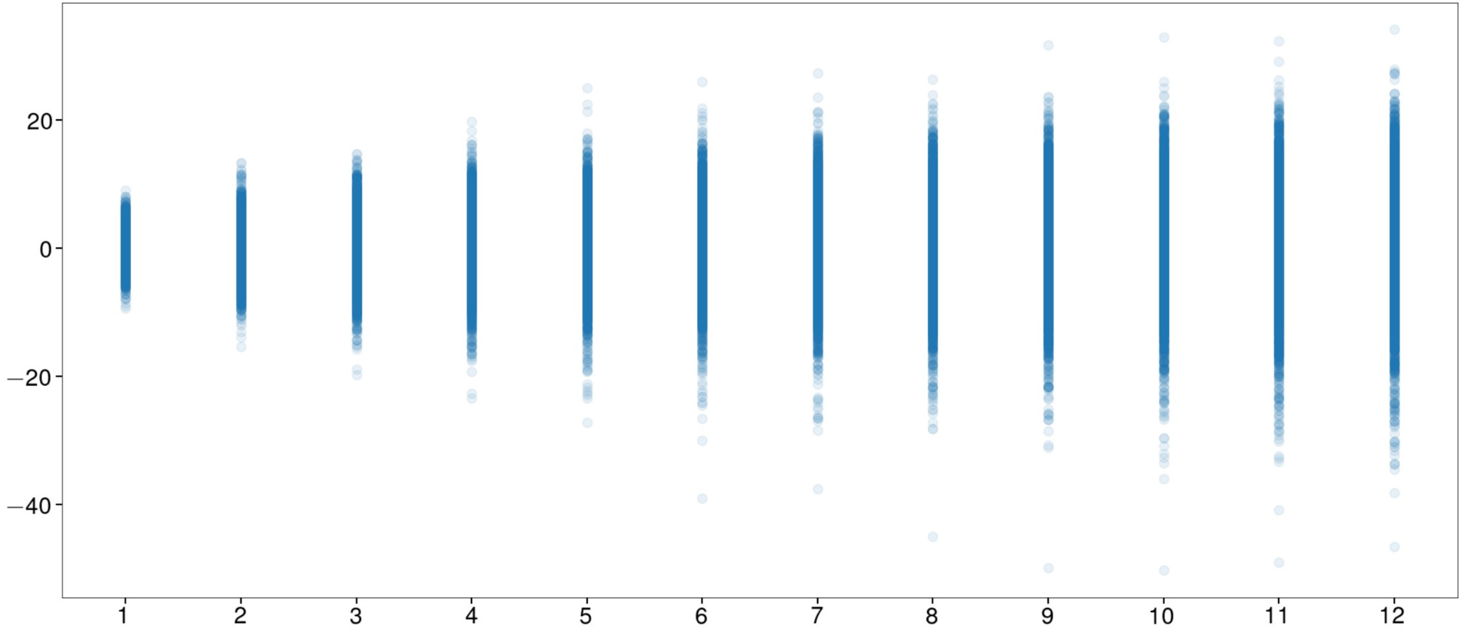
- Densities do not aggregate
- Analytical forecasts are challenging
- Simple solution: simulate

Simulation-based Forecasting

- Condition on final observation
- Simulate using either:
    - Assumed distribution
    - IID Bootstrap of standardized residuals

Normal Simulated Values

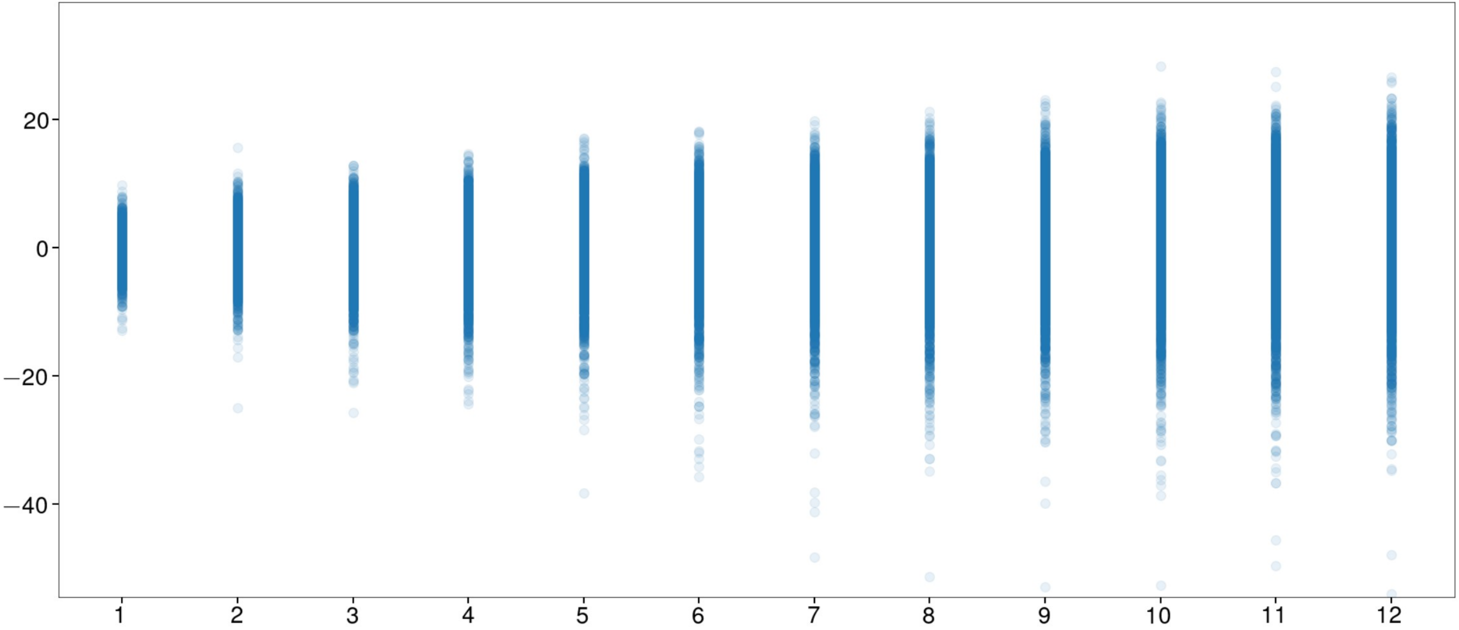
In [33]:
```
plot_sim(normal_fcast)
```



Skew t Simulated Values

In [34]:

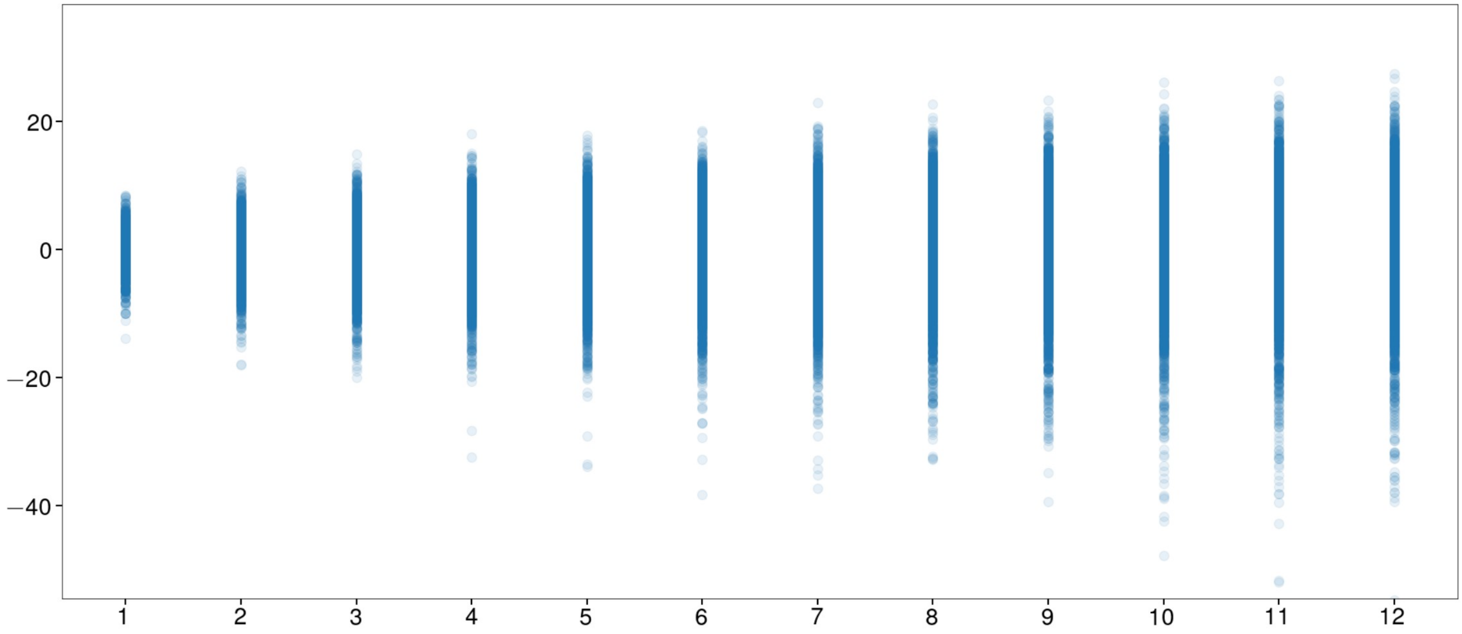
```
plot_sim(skew_t_fcast)
```



Bootstrap Simulated Values

In [35]:

```
plot_sim(bootstrap_fcast)
```

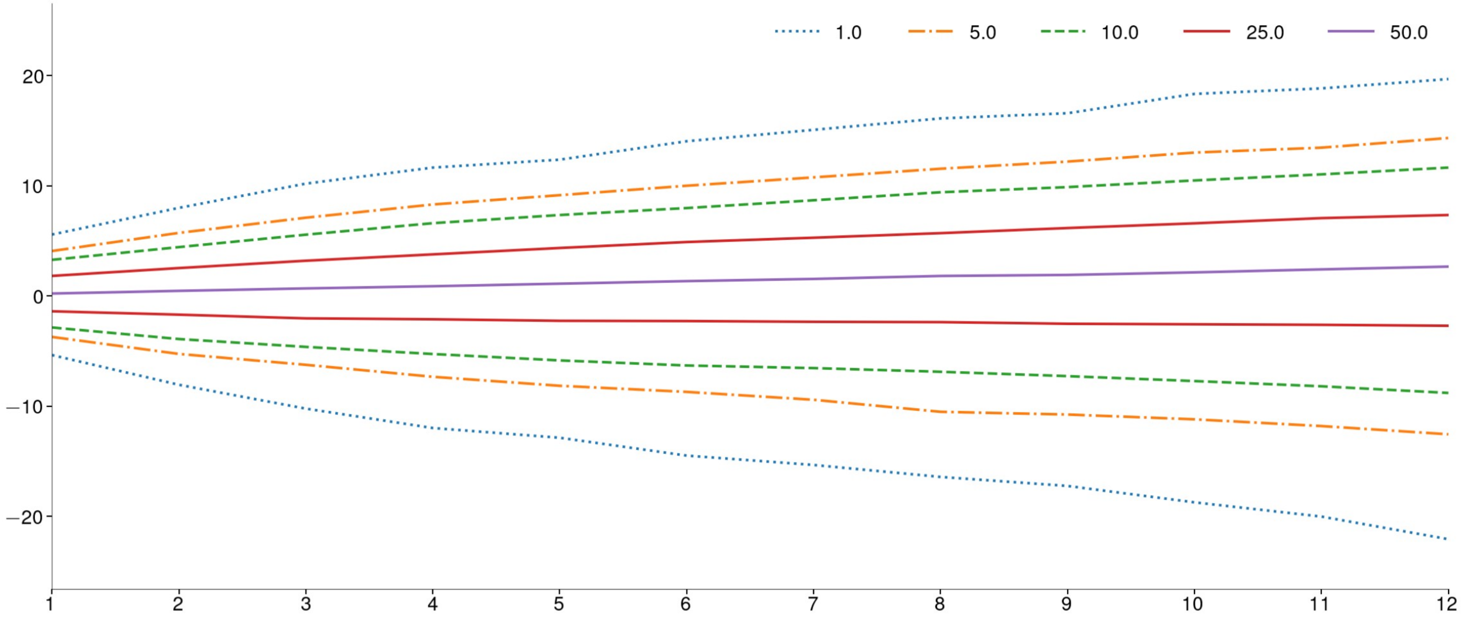


Quantile Plot

- Plot quantiles across horizon
- Simplification of fan plot

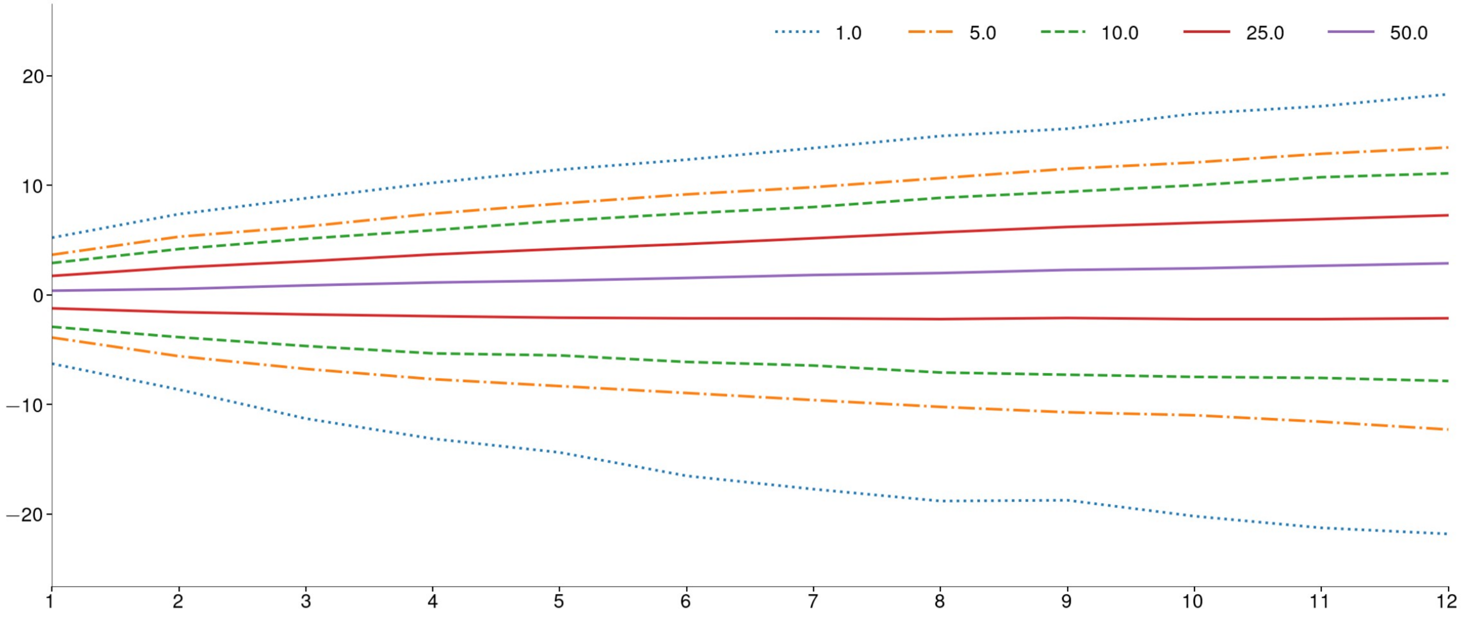
Normal

In [37]:
```
plot_quantile(normal_fcast)
```



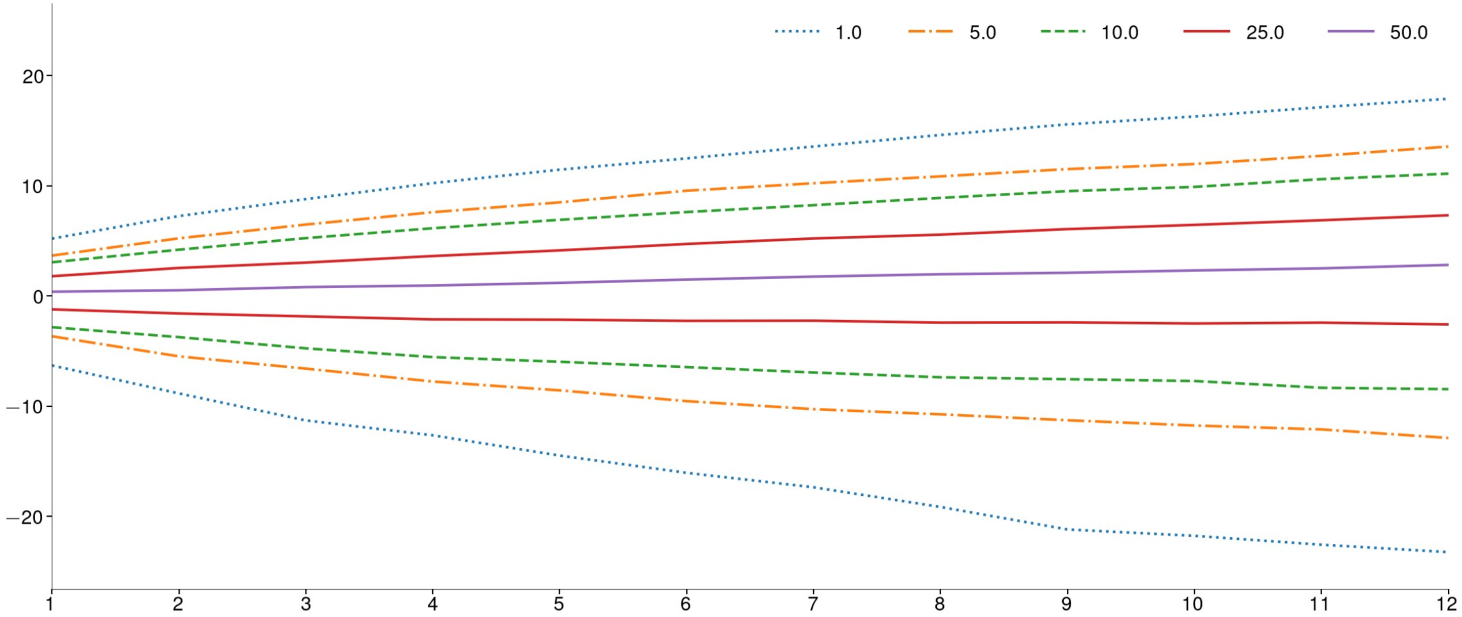
Skew t

In [38]:
```
plot_quantile(skew_t_fcast)
```



Bootstrap

In [39]:
```
plot_quantile(bootstrap_fcast)
```



Quantile-Quantile (QQ) Plots

- Plots the data against a hypothetical distribution

$$\hat{e}_1 < \hat{e}_2 < \ldots < \hat{e}_{N-1} < \hat{e}_N$$

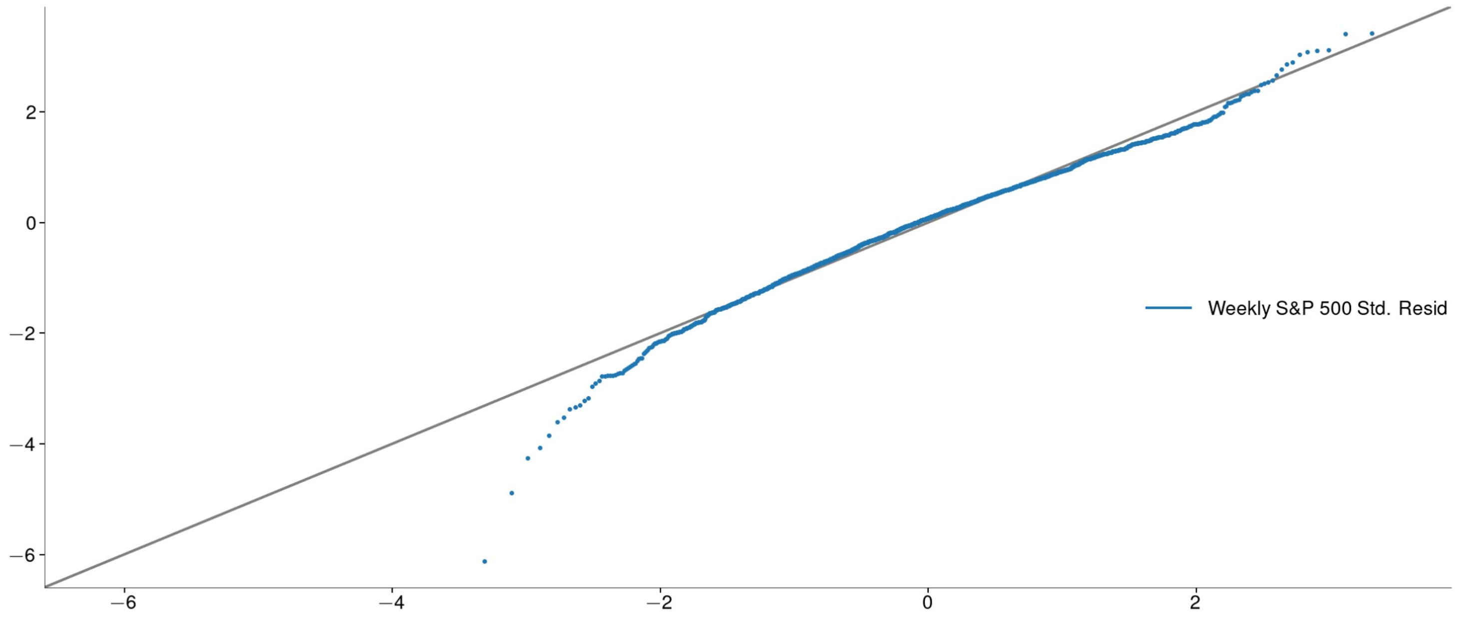
- $N = T$ but used to indicate that the index is not related to time
- e_n against $F^{-1}\left(\frac{j}{T+1}\right)$

$$F^{-1}\left(\frac{1}{T+1}\right) < F^{-1}\left(\frac{2}{T+1}\right) < \ldots < F^{-1}\left(\frac{T-1}{T+1}\right) < F^{-1}\left(\frac{T}{T+1}\right)$$

- F^{-1} is inverse cdf of distribution being used for comparison

Normal

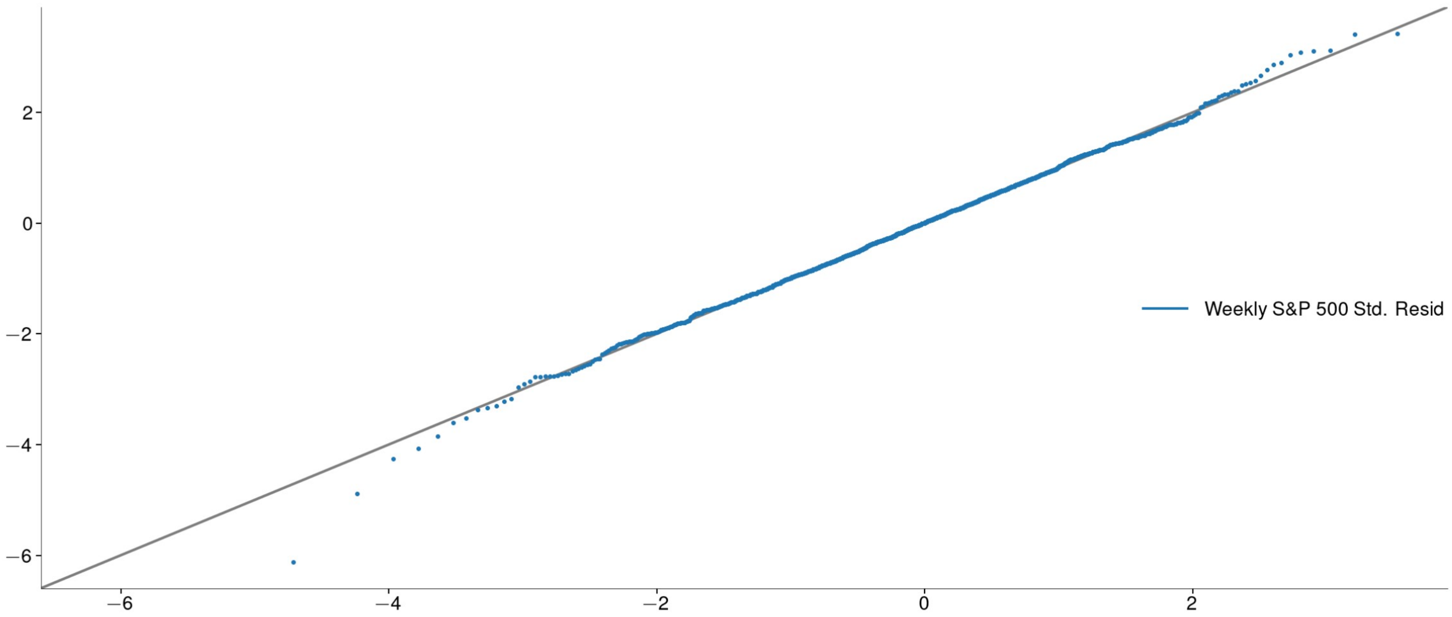
In [41]:
```
qq_normal()
```



Skew t

In [43]:

```
qq_skewt()
```



Kolmogorov-Smirnov Tests

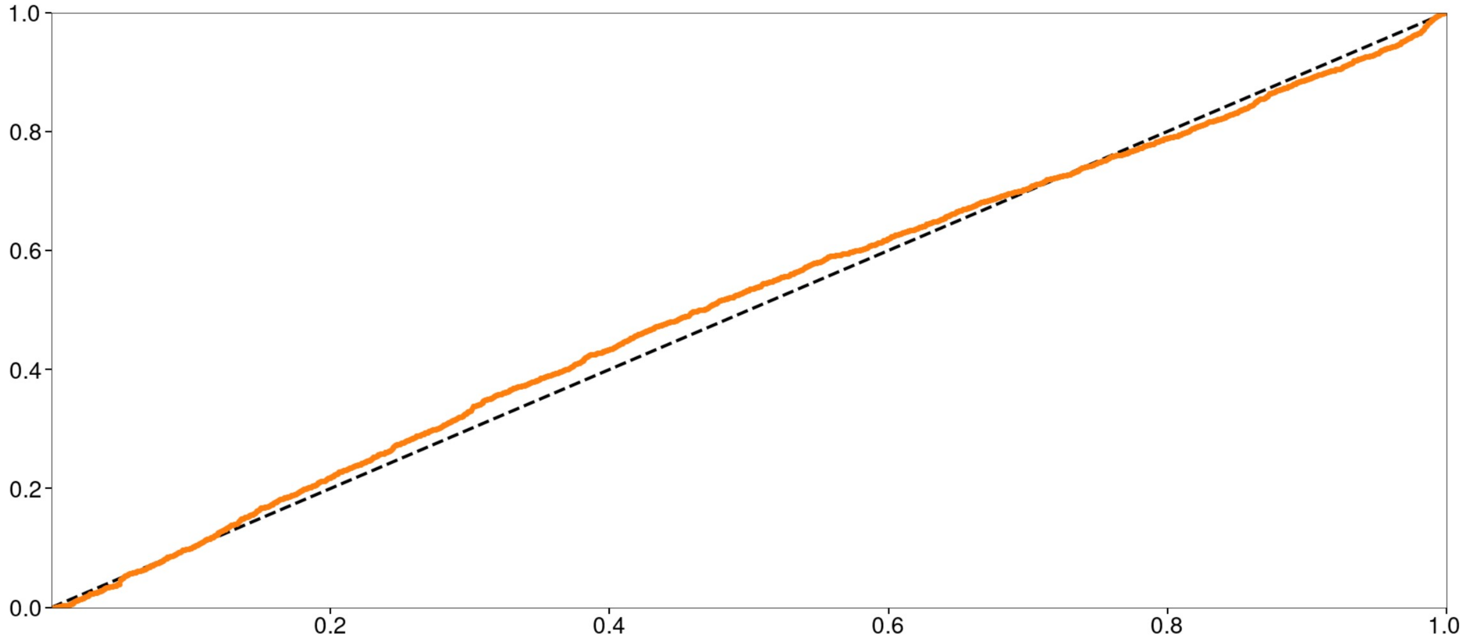
- Formalizes QQ plots
- Key property
    - If $x \sim F$, then $u \equiv F(x) \sim U(0,1)$
    - Test of $U(0,1)$
- KS tests maximum deviation from U(0,1)

$$\max_{\tau} \left| \frac{1}{T} \left(\sum_{i=1}^{\tau} I_{[u_i < \frac{\tau}{T}]} \right) - \frac{\tau}{T} \right|, \quad \tau = 1, 2, \ldots, T$$

- $\frac{1}{T} \sum_{i=1}^{\tau} I_{[u_j < \frac{\tau}{T}]}$ - empirical percentage of u below τ/T
- τ/T - expected fraction below τ/T

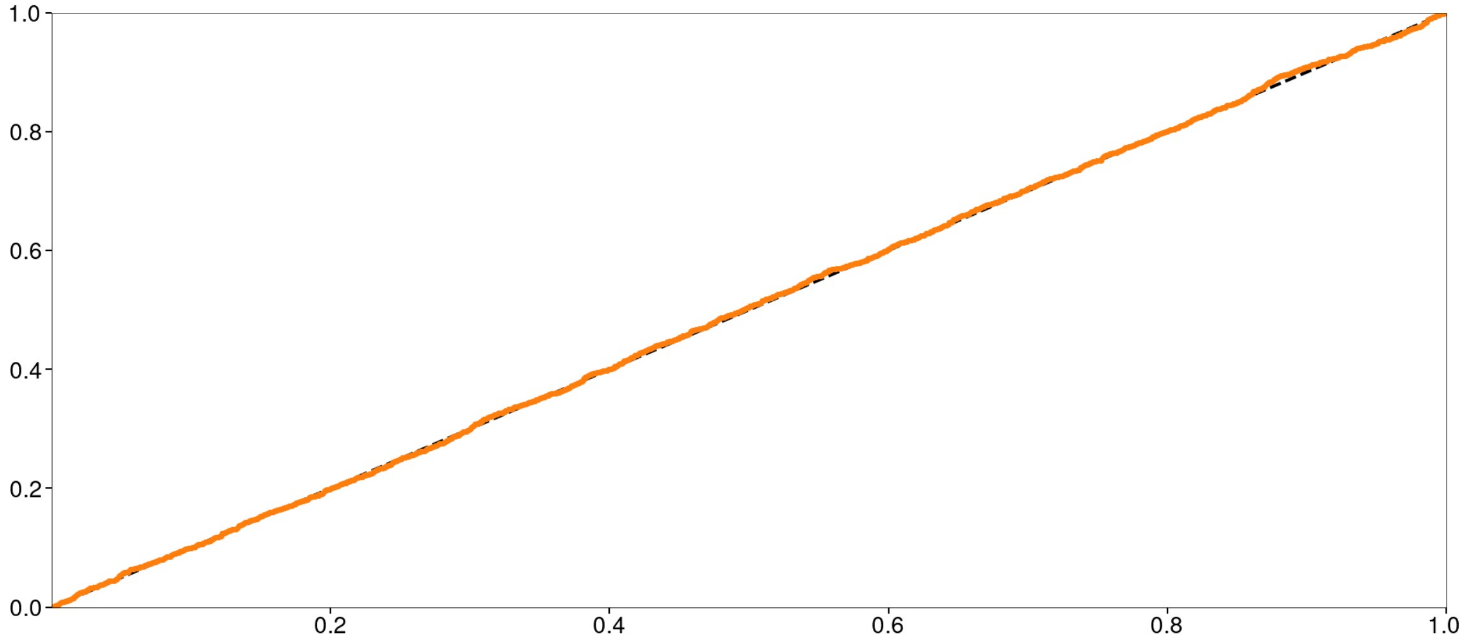
Normal

In [46]:
```
ksplot(u)
```



Skew t

In [48]:
```
ksplot(u)
```



Kolmogorov-Smirnov Tests

In [49]:

```
from scipy.stats import kstest
from statsmodels.stats.diagnostic import kstest_fit
e = tarch_skewt.std_resid
params = tarch_skewt.params
skew_t_dist = tarch_skewt.model.distribution
u = skew_t_dist.cdf(e, params.iloc[-2:])

stat, pval = kstest(u, "uniform")
pretty(f"The KS Stat is {stat:0.4f} and its p-val is {100*pval:0.1f}%")
```

The KS Stat is 0.0126 and its p-val is 87.9%

KS Test with Parameter Estimation Uncertainty

- Model is a complete model so can be easily simulated
- Exact KS distribution tabulated for specific model
- Key idea: compute KS stat on b replications of assumed DGP and use empirical CDF

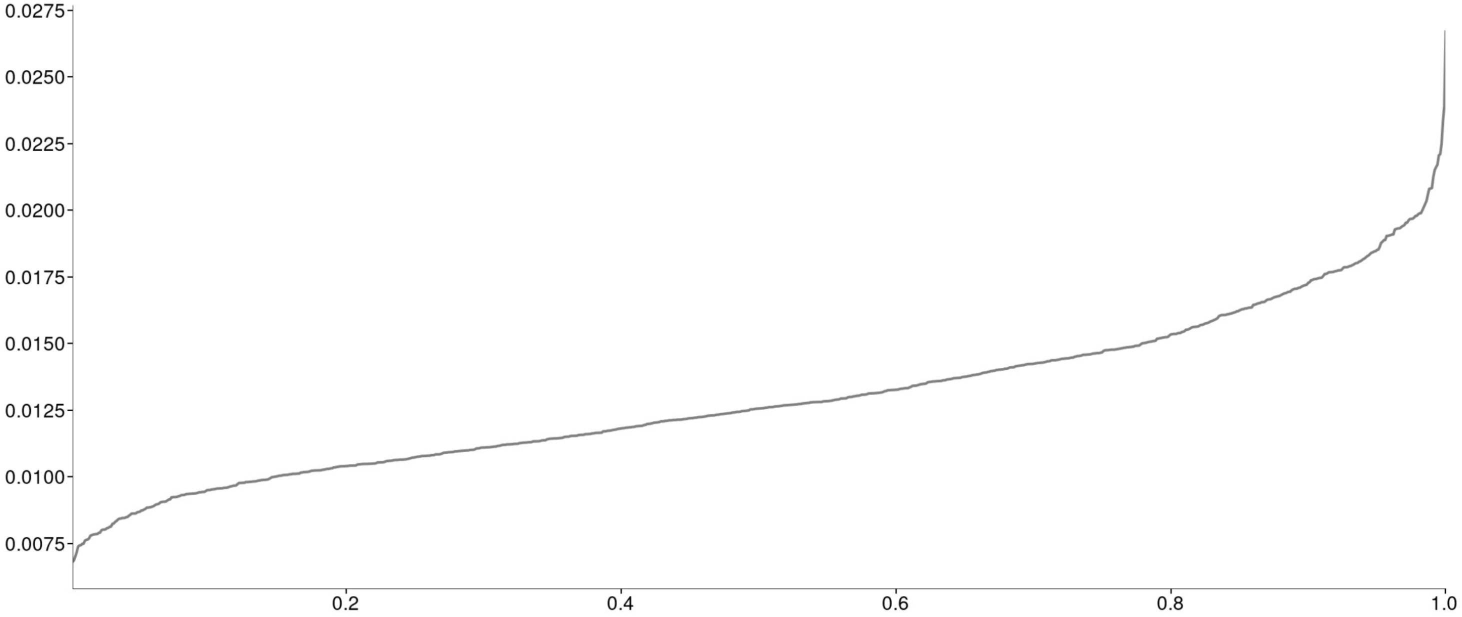
In [51]:

```
pee_pval = (stat < sim_stats).mean()
pretty(f"Original P-value: {100*pval:0.1f}%, Accounting for PEE: {100*pee_pva
l:0.1f}%")
```

Original P-value: 87.9%, Accounting for PEE: 48.9%

Simulated KS Statistics

In [52]:
```
plot(sim_stats)
```



Berkowitz's Test

- Exploits probability integral transform property $\hat{u}_t = F(y_t)$
- Re-transforms the data to a standard normal $\hat{\eta}_t = \Phi^{-1}(\hat{u}_t) = \Phi^{-1}(F(y_t))$

$$\hat{u}_t \overset{iid}{\sim} U(0,1) \Rightarrow \hat{\eta}_t \overset{iid}{\sim} N(0,1)$$

- Test is a likelihood ratio test using an AR(1)

$$\hat{\eta}_t = \phi_0 + \phi_1 \hat{\eta}_{t-1} + \nu_t$$

- Correctly specified

$$H_0 : \phi_0 = 0 \cap \phi_1 = 0 \cap \sigma^2 = \mathrm{V}[\nu_t] = 1$$

- Likelihood ratio

$$2\left(l(\eta_t|\hat{\phi}_0, \hat{\phi}_1, \hat{\sigma}^2) - l(\eta_t|\phi_0 = 0, \phi_1 = 0, \sigma^2 = 1)\right) \sim \chi^2_3$$

Berkowitz Test Results

Skew t

In [54]:

```
summary(ar_res)
```

	coef	std err	z	P>|z|	[0.025	0.975]
intercept	0.0073	0.022	0.339	0.735	-0.035	0.050
ar.L1	-0.0738	0.021	-3.463	0.001	-0.115	-0.032
sigma2	0.9951	0.030	33.004	0.000	0.936	1.054

In [55]:

```
pretty(f"Berkowitz stat: {berkowitz_lr:0.2f}, p-value: {100*berkowitz_pval:0.1f}%")
```

Berkowitz stat: 11.78, p-value: 0.8%

Berkowitz's Test with Parameter Estimation Uncertainty

- Same issues as KS test
- Solution is the same - simulate the empirical CDF of the statistic

In [58]:

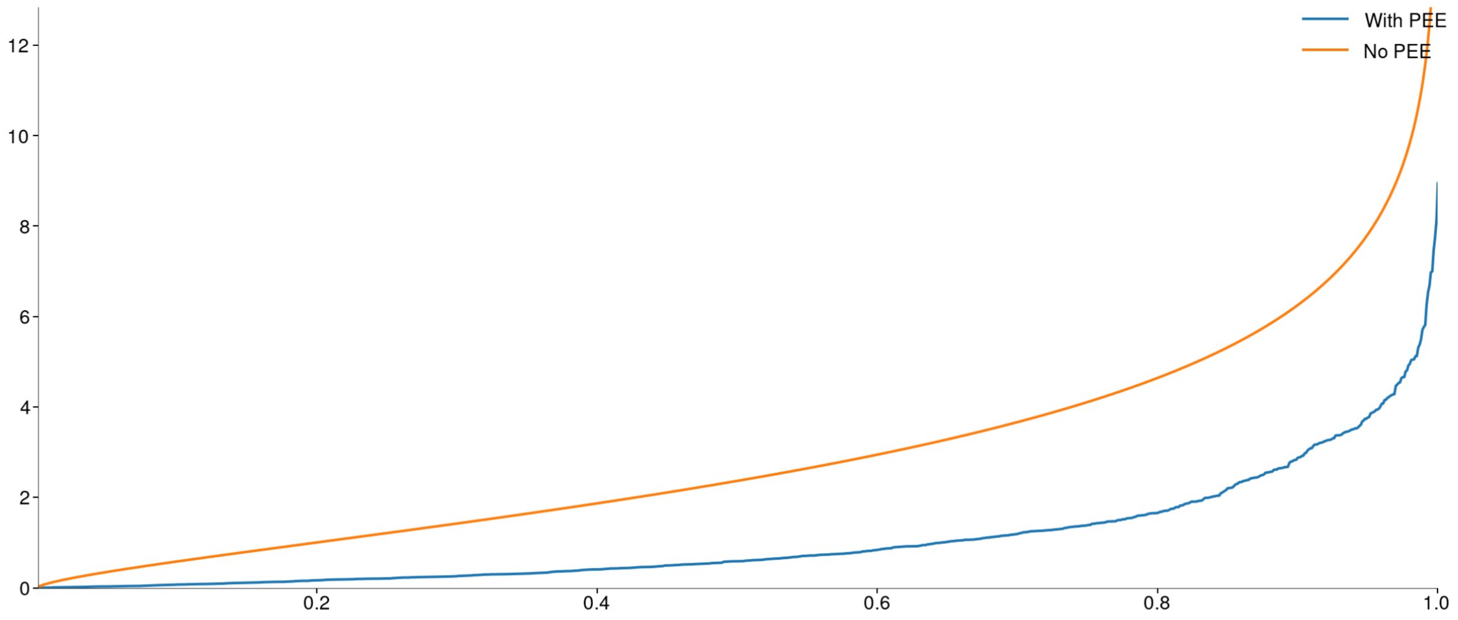
```
berkowitz_pee_pval = (berkowitz_lr < emp_dist).mean()
pretty(
    f"Original P-value: {100*berkowitz_pval:0.1f}%, "
    f"Accounting for PEE: {100*berkowitz_pee_pval:0.1f}%"
)
```

Original P-value: 0.8%, Accounting for PEE: 0.0%

The effect of PEE on the Berkowitz Test

In [60]:

```
plot_berkowitz()
```



Next Week

- Final Classes with Tales
- Office Hours

Schedule TBD

- Revision Class